```
início {método da bolha}
  tipo v = vetor [1:20] inteiro;
  v: VET;
  inteiro: AUX, {auxiliar para troca de elementos}
           BOLHA, {indicador de mais alto elemento fora de ordem}
           LSUP, {indicador do tamanho do vetor a ser pesquisado, sendo o valor
                  inicial igual a 20}
           J; {indicador do elemento do vetor};
  leia (VET);
  LSUP ← 20;
  enquanto LSUP > 1 faça
    BOLHA ← 0;
    para J de 1 até LSUP − 1 faça
      se VET [J] > VET [J + 1]
        então {troca elemento j com j + 1}
          AUX ← VET [J];
          VET [J] ← VET [J + 1];
          VET [J + 1] ← AUX;
          BOLHA ← J;
      fim se;
    fim para
    LSUP ← BOLHA; {aponta para última posição trocada}
  fim enquanto;
  imprima (VET);
fim.
```

Exemplo 9:

Dado um vetor *A* de 128 elementos, verificar se existe um elemento igual a *K* (chave) no vetor. Se existir, imprimir a posição onde foi encontrada a chave; se não, imprimir: "chave *K* não encontrada". O vetor *A* e chave *K* são lidos de cartões.

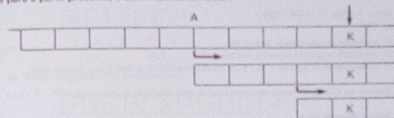
Solução:

```
início
  tipo: v1 = vetor [1:128] inteiro;
  v1: A;
  inteiro: I;
  lógico: ACHOU;
  leia (K);
  leia (A);
  ACHOU ← falso;
  para I de 1 até 128 passo 1 faça
    se A [I] = K
      então imprima (K, "ESTÁ NA POSIÇÃO", I);
            ACHOU ← verdadeiro;
            abandone;
    fim se;
  fim para;
```



```
  se não ACHOU
    então imprima ("A CHAVE", K, "NÃO ESTÁ NO VETOR");
  fim se;
fim.
```

Neste exemplo, a pesquisa a um elemento ficará mais eficiente se utilizarmos a pesquisa binária, desde que o vetor já esteja ordenado. Nesta pesquisa procuramos o elemento *K* dividindo o vetor em duas partes e testando em qual das duas ele deveria estar. Procedendo da mesma forma para a parte provável, e assim sucessivamente:




```
início {pesquisa binária}
  inteiro: COMEÇO, {indicador do primeiro elemento da parte do vetor
                    a considerar}
           FIM, {indicador do último elemento da parte do vetor a
                 considerar}
           MEIO, {indicador do elemento do meio da parte do vetor considerada}
           K, {elemento procurado}
  tipo v = vetor [1:128] inteiro;
  v: A;
  leia: (A, K);
  COMEÇO ← 1;
  FIM ← 128;
  repita
    MEIO ← (COMEÇO + FIM) /2;
    se K < A [MEIO]
      então FIM ← MEIO − 1;
      senão COMEÇO ← MEIO + 1;
    fim se
  até A [MEIO] = K ou COMEÇO > FIM;
  se A [MEIO] ≠ K
    então imprima ("NÃO EXISTE O ELEMENTO");
    senão imprima ("ESTÁ NA POSIÇÃO:", MEIO);
  fim se;
fim.
```

Na pesquisa seqüencial simples, o *número médio* de comparações que devem ser feitas até encontrar a chave é N/2, sendo *N* o número de elementos do vetor.

No nosso caso, teríamos em média, 128/2 = 64 comparações.

Na pesquisa binária, o *número máximo* de comparações é $\log_2 N$.

No nosso caso, teríamos $\log_2 128 = 7$ comparações, no máximo.